EECS3311 Software Design
Fall 2017
Exercise: Proving Correctness of Loops

Chen-Wei Wang

Consider the following query which involves the use of a loop to find the maximum value from an integer array:

```
1  find_max (a: ARRAY [INTEGER]): INTEGER
2      require
3          not_empty: a.count > 0
4      local
5          i: INTEGER
6      do
7          from
8              i := a.lower
9              Result := a [i]
10         invariant
11             -- Predicate Equivalent: ∀j | a.lower ≤ j < i • Result ≥ a[j]
12             across
13                 a.lower |..| (i − 1) as j
14             all
15                 Result >= a [j.item]
16             end
17         until
18             i > a.upper
19         loop
20             if a [i] > Result then
21                 Result := a [i]
22             end
23             i := i + 1
24         variant
25             a.upper − i + 1
26         end
27     ensure
28         -- Predicate Equivalent: ∀j | a.lower ≤ j ≤ a.upper • Result ≥ a[j]
29         across
30             a.lower |..| a.upper as j
31         all
32             Result >= a [j.item]
33         end
34     end
```

Your Tasks

Prove or disprove that the above program is *totally* correct, which involves the following steps:

1. State formally, in terms of Hoare Triples, the obligations for proving that:
   - The loop is *partially* correct (without considering termination); and
   - The loop terminates.
2. For each of the Hoare triple $\{\ Q\ \}\ S\ \{\ R\ \}$ in Step 1, calculate the corresponding weakest precondition (i.e., $wp\ (S,\ R)$).
3. Prove or disprove that the calculated wp is equal to or weaker than the corresponding precondition (i.e., prove or disprove that $Q \Rightarrow wp\ (S,\ R)$).

1 Partial Correctness

1.1 Establishing the Loop Invariant

Proof Obligation:

$$\{\ a.count > 0\ \}$$
```
    i := a.lower; Result := a[i]
```
$$\{\ \forall j \mid a.lower \le j \le i-1 \bullet \boxed{a.lower \le j \wedge j \le a.upper} \wedge \mathbf{Result} \ge a[j]\ \}$$

Notice that the augmented constraint $\boxed{a.lower \le j \wedge j \le a.upper}$ is due to the array indexing expression `a[j]`. Similar augmentation is performed for each occurrence of an array indexing expression.

1.2 Maintaining the Loop Invariant

Proof Obligation:

$$\{\ \neg(i > a.upper) \wedge (\ \forall j \mid a.lower \le j \le i-1 \bullet \boxed{a.lower \le j \wedge j \le a.upper} \wedge \mathbf{Result} \ge a[j]\ )\ \}$$
```
    if a[i] > Result then Result := a[i] end; i := i + 1
```
$$\{\ \forall j \mid a.lower \le j \le i-1 \bullet \boxed{a.lower \le j \wedge j \le a.upper} \wedge \mathbf{Result} \ge a[j]\ \}$$

1.3 Establishing the Postcondition

Proof Obligation:

$$(i > a.upper) \wedge (\ \forall j \mid a.lower \le j \le i-1 \bullet \boxed{a.lower \le j \wedge j \le a.upper} \wedge \mathbf{Result} \ge a[j]\ )$$
$$\Rightarrow (\ \forall j \mid a.lower \le j \le a.upper \bullet \boxed{a.lower \le j \wedge j \le a.upper} \wedge \mathbf{Result} \ge a[j]\ )$$

2 Termination

2.1 Loop Variant Stays Positive

Proof Obligation:

$$\{\ \neg(i > a.upper) \wedge (\ \forall j \mid a.lower \le j \le i-1 \bullet \boxed{a.lower \le j \wedge j \le a.upper} \wedge \mathbf{Result} \ge a[j]\ )\ \}$$
```
    if a[i] > Result then Result := a[i] end; i := i + 1
```
$$\{\ a.upper - i + 1 > 0\ \}$$

2.2 Loop Variant Decreases

Proof Obligation:

$$\{\ \neg(i > a.upper) \wedge (\ \forall j \mid a.lower \le j \le i-1 \bullet \boxed{a.lower \le j \wedge j \le a.upper} \wedge \mathbf{Result} \ge a[j]\ )\ \}$$
```
    if a[i] > Result then Result := a[i] end; i := i + 1
```
$$\{\ a.upper - i + 1 < a.upper_0 - i_0 + 1\ \}$$

Solution to Proving (1.2)

We first calculate the wp for the loop body to maintain the LI:

$$
\begin{array}{ll}
 & wp(\texttt{if a[i] > Result then Result := a[i] end; i := i + 1}, \boxed{\forall j \,|\, a.lower \le j \le i-1 \bullet a.lower \le j \wedge j \le a.upper \wedge \textbf{Result} \ge a[j]}) \\
= & \{wp \texttt{ rule for seq. comp. }\} \\
 & wp(\texttt{if a[i] > Result then Result := a[i] end}, \boxed{wp(\texttt{i := i + 1}, \boxed{\forall j \,|\, a.lower \le j \le i-1 \bullet a.lower \le j \wedge j \le a.upper \wedge \textbf{Result} \ge a[j]})}) \\
= & \{wp \texttt{ rule for assignment}\} \\
 & wp(\texttt{if a[i] > Result then Result := a[i] end}, \boxed{\forall j \,|\, a.lower \le j \le \textbf{i} \bullet a.lower \le j \wedge j \le a.upper \wedge \textbf{Result} \ge a[j]}) \\
= & \{wp \texttt{ rule for conditional}\} \\
 & a[i] > \textbf{Result} \implies wp(\texttt{Result := a[i]}, \boxed{\forall j \,|\, a.lower \le j \le i \bullet a.lower \le j \wedge j \le a.upper \wedge \textbf{Result} \ge a[j]}) \\
 & \wedge \\
 & a[i] \le \textbf{Result} \implies wp(\texttt{Result := Result}, \boxed{\forall j \,|\, a.lower \le j \le i \bullet a.lower \le j \wedge j \le a.upper \wedge \textbf{Result} \ge a[j]}) \\
= & \{wp \texttt{ rule for assignment, twice}\} \\
 & a[i] > \textbf{Result} \implies \forall j \,|\, a.lower \le j \le i \bullet a.lower \le j \wedge j \le a.upper \wedge \textbf{a[i]} \ge a[j] \\
 & \wedge \\
 & a[i] \le \textbf{Result} \implies \forall j \,|\, a.lower \le j \le i \bullet a.lower \le j \wedge j \le a.upper \wedge \textbf{Result} \ge a[j]
\end{array}
$$

We then prove that the precondition (i.e., ¬(exit condition) and LI) is no weaker than the above calculated wp:

- To prove:

$$
\begin{array}{l}
\neg(i > a.upper) \wedge (\ \forall j \,|\, a.lower \le j \le i-1 \bullet a.lower \le j \wedge j \le a.upper \wedge \textbf{Result} \ge a[j]\ ) \\
\quad \implies a[i] > \textbf{Result} \implies \boxed{\forall j \,|\, a.lower \le j \le i \bullet a.lower \le j \wedge j \le a.upper \wedge a[i] \ge a[j]}
\end{array}
$$

  Proof:

$$
\begin{array}{ll}
 & \boxed{\forall j \,|\, a.lower \le j \le i \bullet a.lower \le j \wedge j \le a.upper \wedge a[i] \ge a[j]} \\
\equiv & \{\texttt{split range: } \forall j \,|\, a.lower \le j \le i \bullet P(j) \equiv (\forall j \,|\, a.lower \le j \le i-1) \wedge P(i)\} \\
 & (\forall j \,|\, a.lower \le j \le \textbf{i - 1} \bullet a.lower \le j \wedge j \le a.upper \wedge a[i] \ge a[j]) \wedge (a.lower \le \textbf{i} \wedge \textbf{i} \le a.upper \wedge a[i] \ge a[\textbf{i}]) \\
\equiv & \{\texttt{antecedent: } a[i] > \texttt{Result}\texttt{; and RHS of precond: } \forall j \,|\, a.lower \le j \le i-1 \bullet a.lower \le j \wedge j \le a.upper \wedge \texttt{Result} \ge a[j]\} \\
 & true \wedge (a.lower \le i \wedge i \le a.upper \wedge a[i] \ge a[i]) \\
\equiv & \{\texttt{LHS of precond: } \neg(i > a.upper) \texttt{ and } a[i] \ge a[i] \equiv true\} \\
 & true
\end{array}
$$

- To prove:

$$
\begin{array}{l}
\neg(i > a.upper) \wedge (\ \forall j \,|\, a.lower \le j \le i-1 \bullet a.lower \le j \wedge j \le a.upper \wedge \textbf{Result} \ge a[j]\ ) \\
\quad \implies a[i] \le \textbf{Result} \implies \forall j \,|\, a.lower \le j \le i \bullet a.lower \le j \wedge j \le a.upper \wedge \textbf{Result} \ge a[j]
\end{array}
$$

  (*Exercise*)